Instacart Shopping Data
Final Report
Sava Dashev

Problem statement

The online shopping got firm grip on customers. All type of products are offered. In the last few years, we have large increase on groceries that are offered online. Many stores support ordering online and picking at store.

How do the stores prepare to better serve their customers? That is the goal. What products do customers order often? Which products are bought together? How often do the customers come back?

These questions help stores to serve better customers. There is more. How can we divide customers into groups related to the behavior they present online? Are there preferences to a particular day, or particular hour? Do the customers coming more often buy the same products as these coming once a month?

Dividing customers in meaningful groups helps predicting what these customers may order. Also, with segmentation we may create incentives targeting that segment based on what these customers previously ordered.

This survey will try to use unsupervised learning to cluster customers into groups. The grocery retailer may want to use segmentation to target with ads different groups of customers and create incentives for these customers to come back.

Also, segmentation may help with planning inventory and cut the costs for the company. This will be done by predicting how much of each product may be sold.

The intended customer for this project is a retailer, which can manage or change preferences to save money by planning and predicting the needs of customers based on past experience. The type of goods offered is food, which one needs to replenish from time to time or more often.

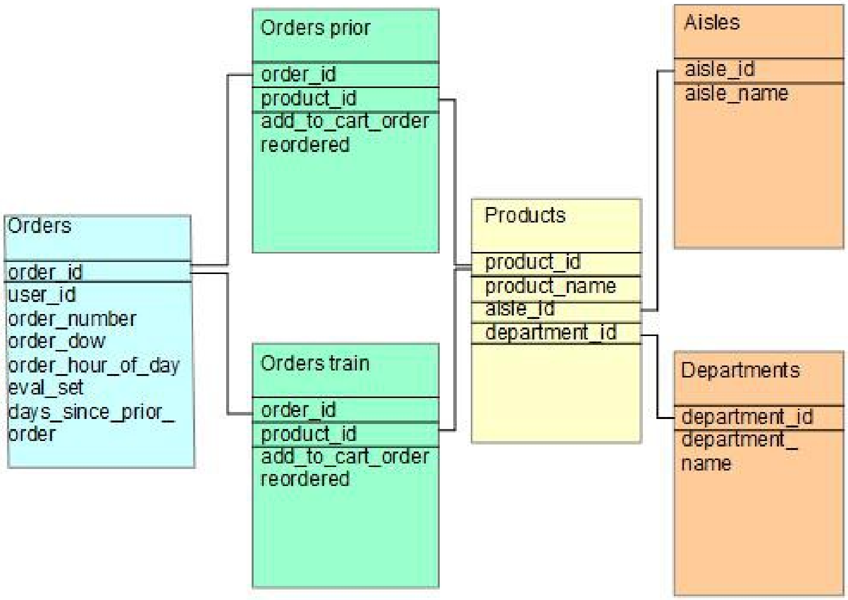
Dataset

The data is collected in 2017. It is from old Kaggle competition. The zip file contains 5 files with separated data for transactions.

The main file contains information about each order. In the file, each row is data for one order. The values in the row are user ID, the date and time of the order, how many days elapsed since the prior order. Each order has order ID. We also have counter variable, how many orders each customer has placed.

The orders are divided in three groups. The largest group is called “prior”. These are all orders, for each customer, but the last order. The last order, for each customer are separated into two groups. One of the groups is called “train”, and the other group is called “test”. The train group consists of last order for about ⅔ of all of the users. This last order will be used to train model to predict items in that order. The rest of the last orders for the remainder of the users will be used for testing the model in the competition. These orders do not have data available for the items in them.

The user id-s are randomized. There is no data how much each item cost, or from which store it was ordered.



Data structure

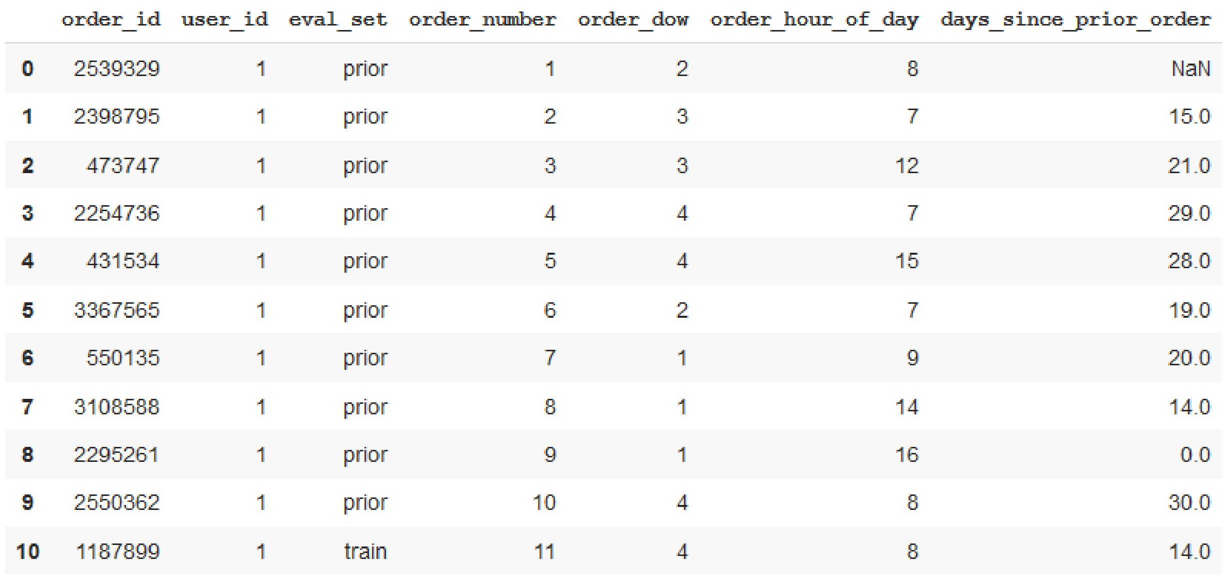
Two other files contain the data for each order. One of the files contain “prior” items order and the other contain “train” items. Each row in this table has values connecting order ID and product ID. The other bits of information for this table is “reordered” - the value of 0 shows that this is the first time the user placing order for that item, and 1 otherwise. The last entry is a number-counter is the item 1st, 2nd, 3rd … in the basket for that order.

The description of each product is distributed in three files. The first file contains the name of each product and two other values: aisle ID, and department ID. The last two tables, very small, give each aisle ID name and each department ID name.

Cleaning data

The set of data was relatively clean.

We read each file into separate dataframe. We looked for missing values. For each numerical variable, we got the same number of entries, as the number of rows. The only exception was the variable showing days since prior order. For each user, the first order has NaN value, nothing to start counting from. The data had to use NaN, sometimes, there were two orders in the same day!



	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0
5	3367565	1	prior	6	2	7	19.0
6	550135	1	prior	7	1	9	20.0
7	3108588	1	prior	8	1	14	14.0
8	2295261	1	prior	9	1	16	0.0
9	2550362	1	prior	10	4	8	30.0
10	1187899	1	train	11	4	8	14.0

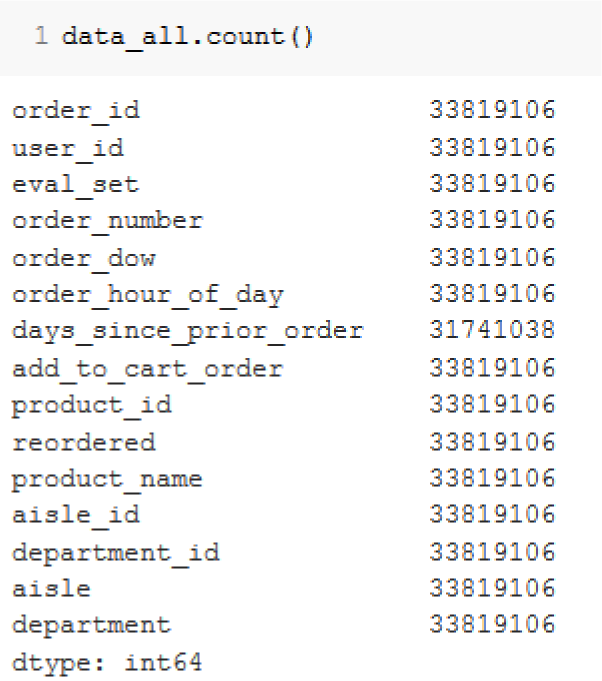
Orders dataframe head

We also checked for duplicated orders. All orders have unique numbers, no duplicates for train, test and prior. Each order belongs to exactly ONE set.

After checking data, we combined data into one dataframe for EDA and further processing. First, we combined prior and train dataframes. These two dataframes have the same type of data. Next, we combined prior and train data with orders dataframe.

We collected products, aisles and departments into one dataframe containing the ID-s and the name of each product, department and aisle.

The last step was to combine the orders data frame and the products data frames.



```
data_all.count()
```

```
order_id                  33819106
user_id                   33819106
eval_set                  33819106
order_number              33819106
order_dow                 33819106
order_hour_of_day         33819106
days_since_prior_order    31741038
add_to_cart_order         33819106
product_id                33819106
reordered                 33819106
product_name              33819106
aisle_id                  33819106
department_id             33819106
aisle                     33819106
department                33819106
dtype: int64
```

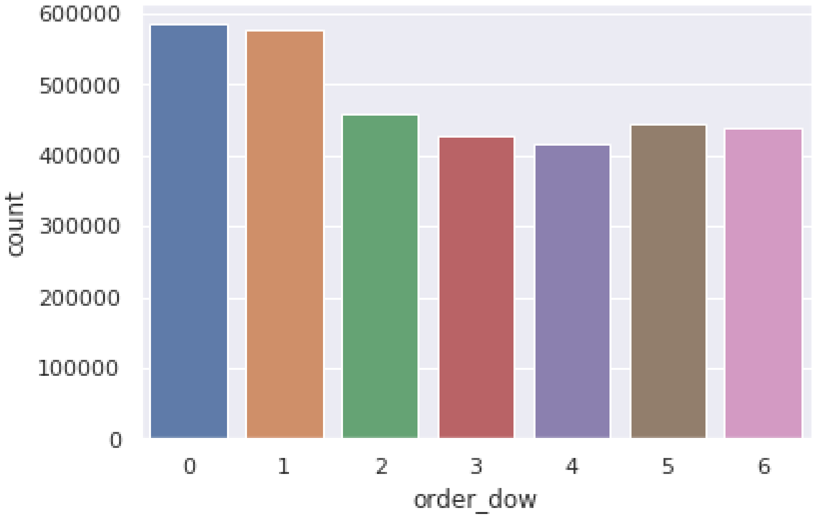
Combined dataframe variables count

All variable in the combined dataframe have the same number of entries, with exception of days since prior order. The first order have no data for that variable, because it is the initial one. The zero is reserved for two orders in the same day.

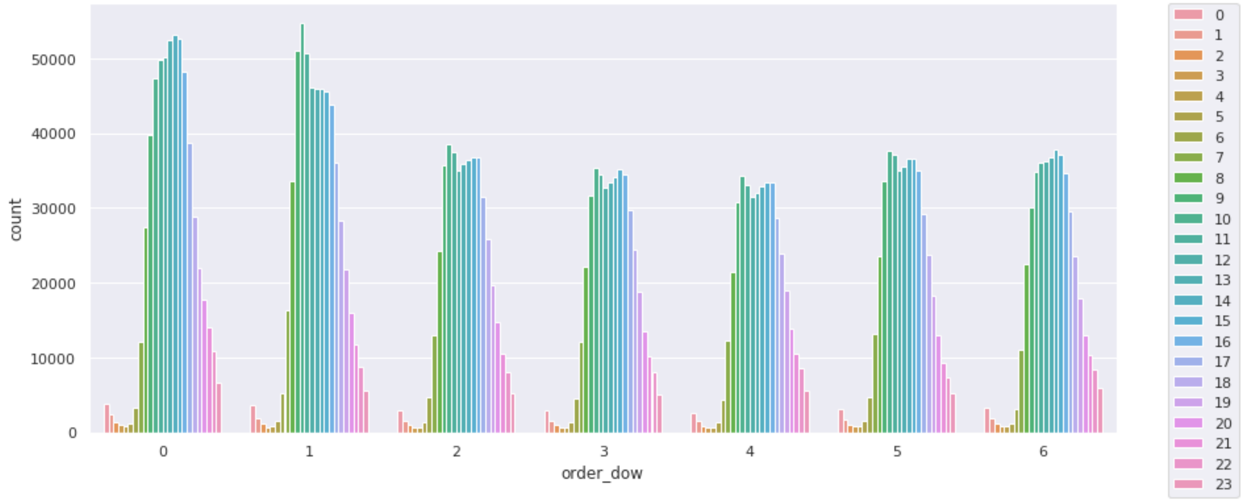
When we finished, we checked the combined data and saved it into a large file.

Exploratory analysis

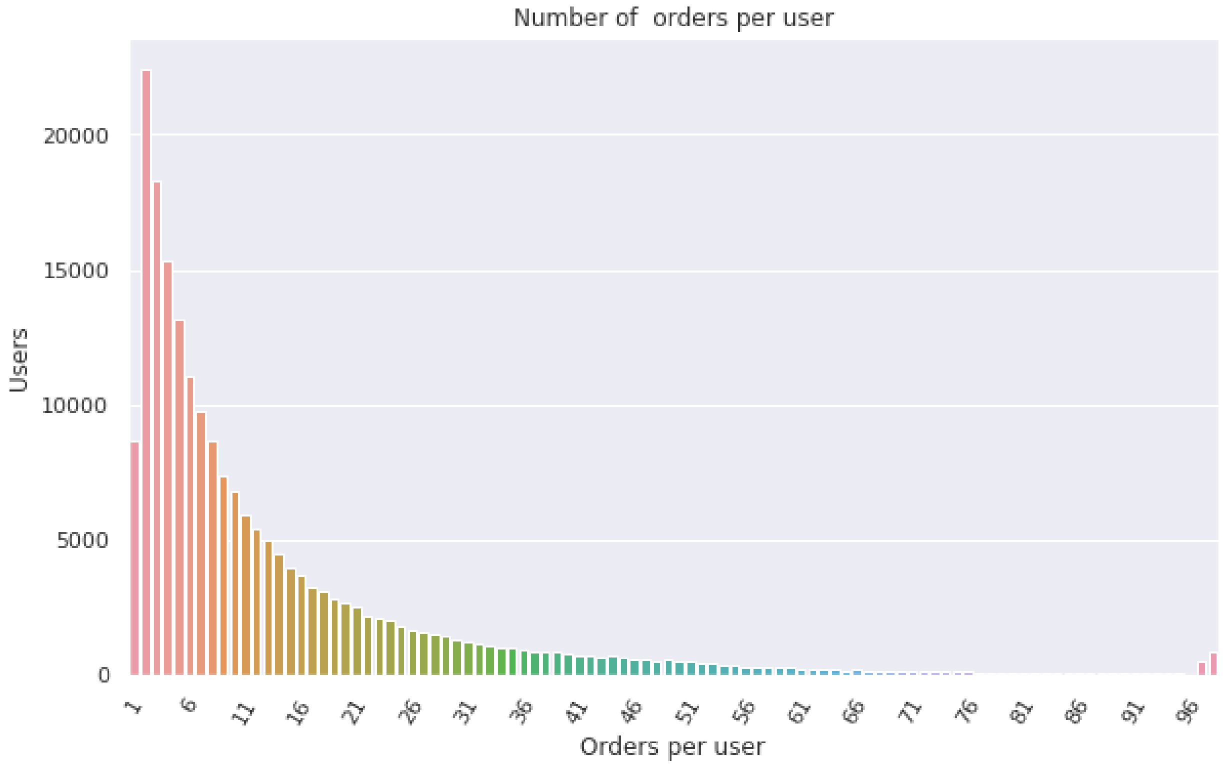
The distribution of orders over time follows two different patterns. Users are most active ordering Saturday and Sunday. Least active day is Wednesday.



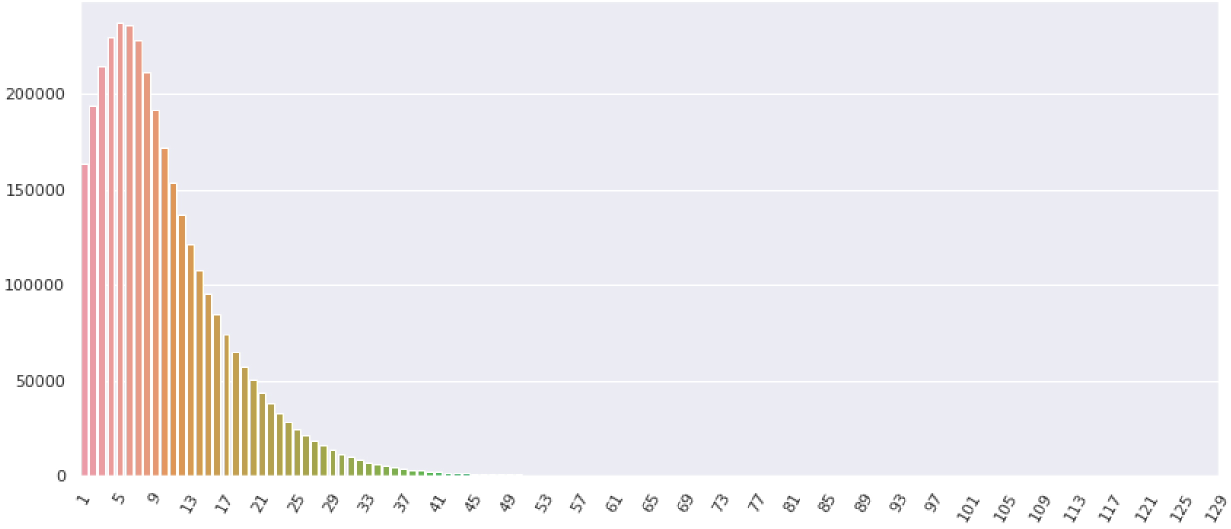
For each hour of day, Monday to Thursday the count of orders each day follows similar distribution. Users are most active around 10:00 am and 4:00 pm. Friday, Saturday and Sunday have only one local maximum of order counts.



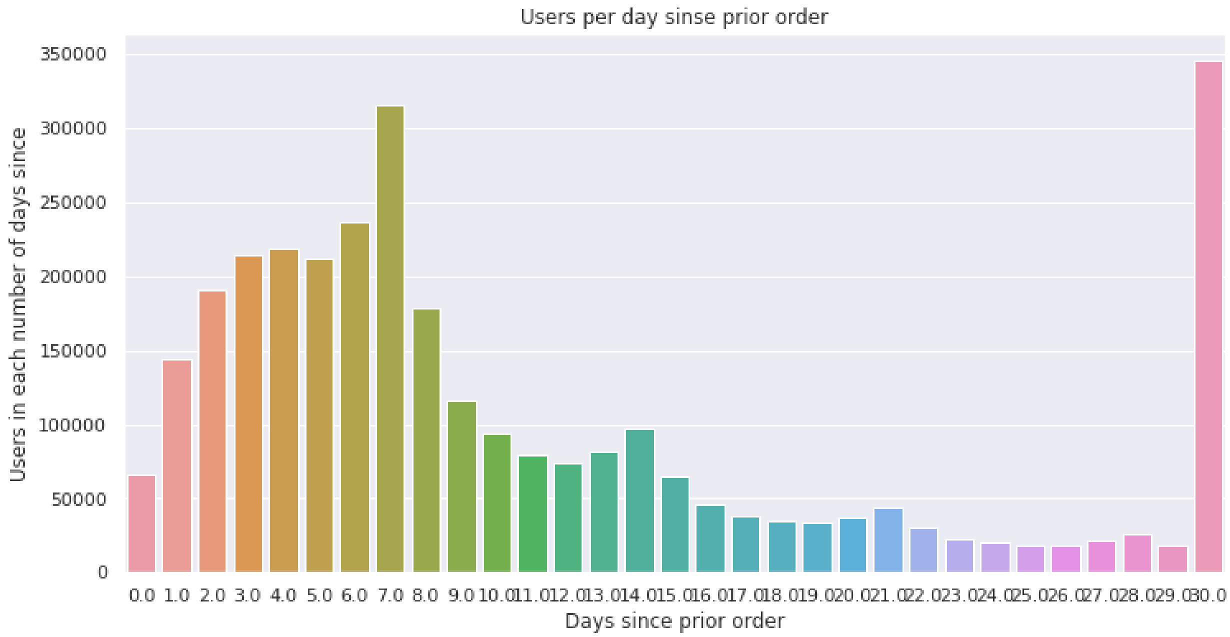
The number of orders per user is between 4 and 100. The most users have placed 4 orders.



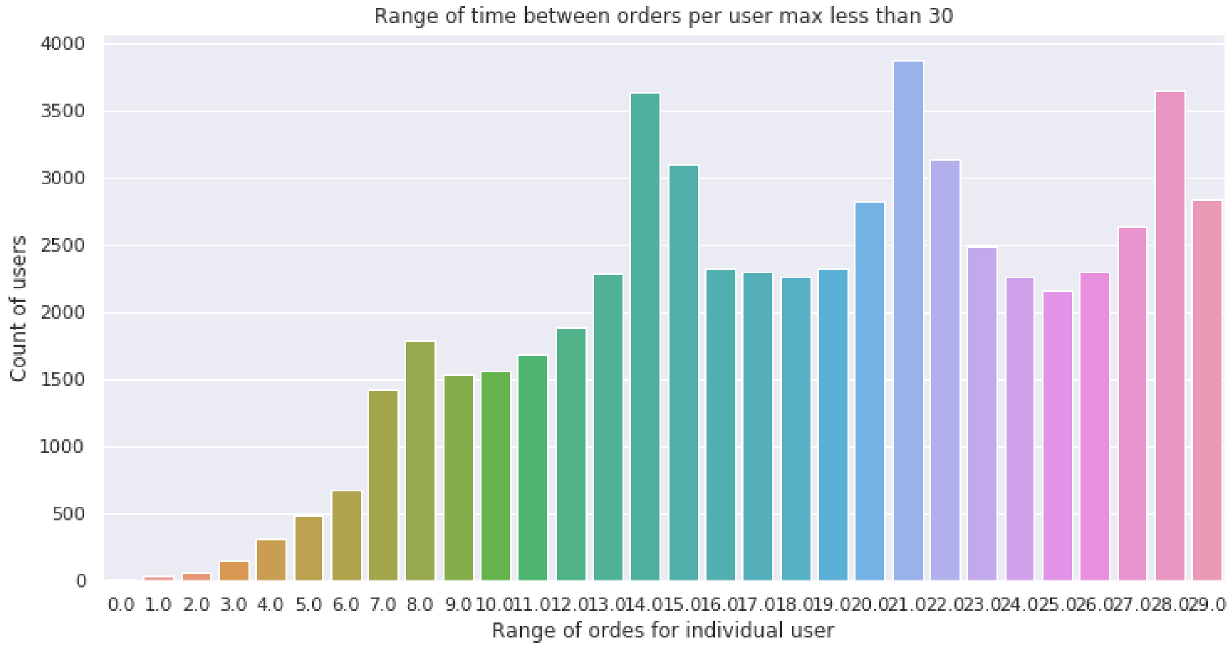
Most users placed small orders. The mode of number of items is 5. The largest order contain about 120 items. The distribution for number of items in order is right tail distribution.



We look how often users shop for food. Grouping by days since prior order, we see picks of activities around 7, 14, 21 and 28 days.

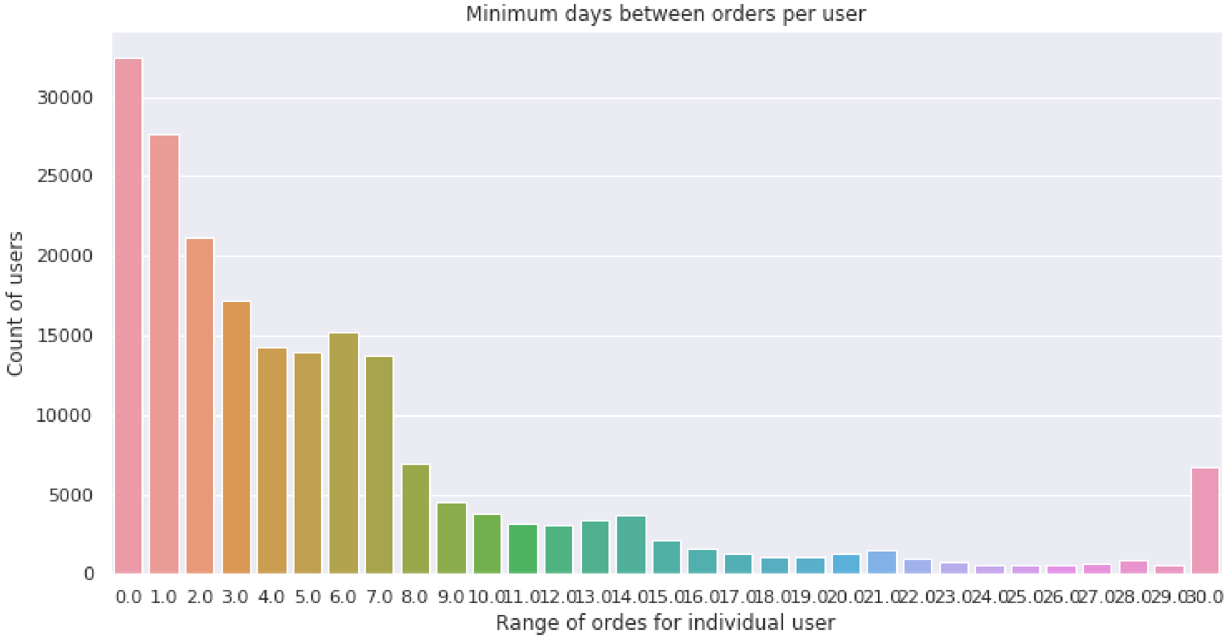


For some users, they place orders after more than 30 days. The data shows that we have 30 or more days since prior order.



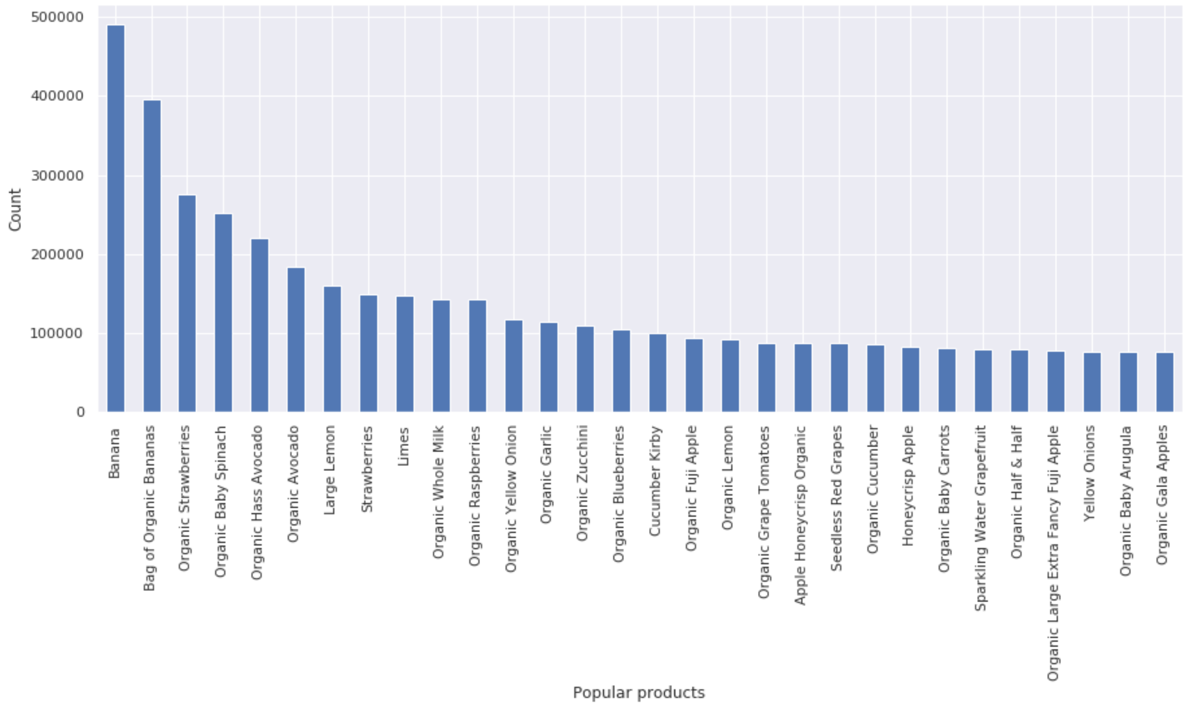
The maximum number of days between orders for some users is 30 days.

For another group of users, the number of days between orders is no more than 29 days. We have local pics around 7, 14 and 21 days. The most users have 21 days since prior order as max number of days between orders.



We can also see the least number of days between orders. Some 30,000 users shopped more than once in the same day. And another group of 7,500 users, shop online no more often than once a month.

What items are most often ordered? Bananas and organic bananas are the first two items. In the first 30 most ordered items, more than half are organic.



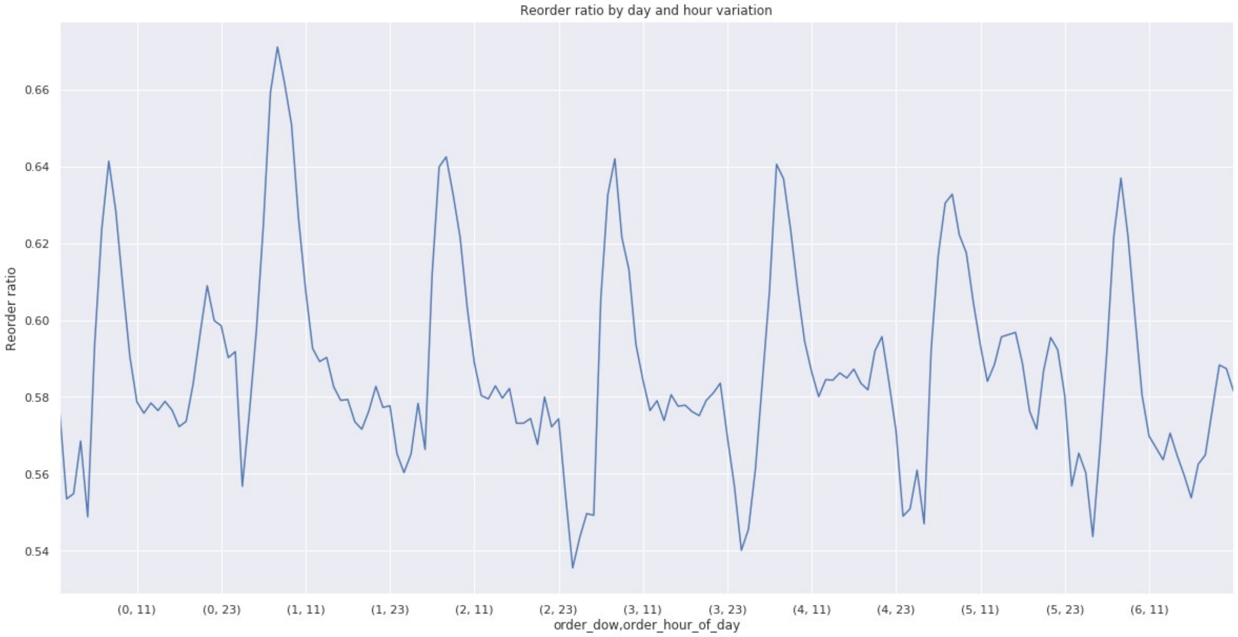
We also looked into most popular departments and aisles. The most popular aisles are fresh vegetables and fresh fruits, for the departments - diary eggs. The least popular is department is bulk.

aisle	
Fresh fruits	3792661
Fresh vegetables	3568630
Packaged vegetables fruits	1843806
Yogurt	1507583
Packaged cheese	1021462
Milk	923659
Water seltzer sparkling water	878150
Chips pretzels	753739
Soy lactose free	664493
Bread	608469

Statistical inferences

Using our data, we can calculate reorder ratio.

The reorder ratio varies with time. Grouping items by day and time of day, the reorder ratio varies from 0.67 to 0.54.



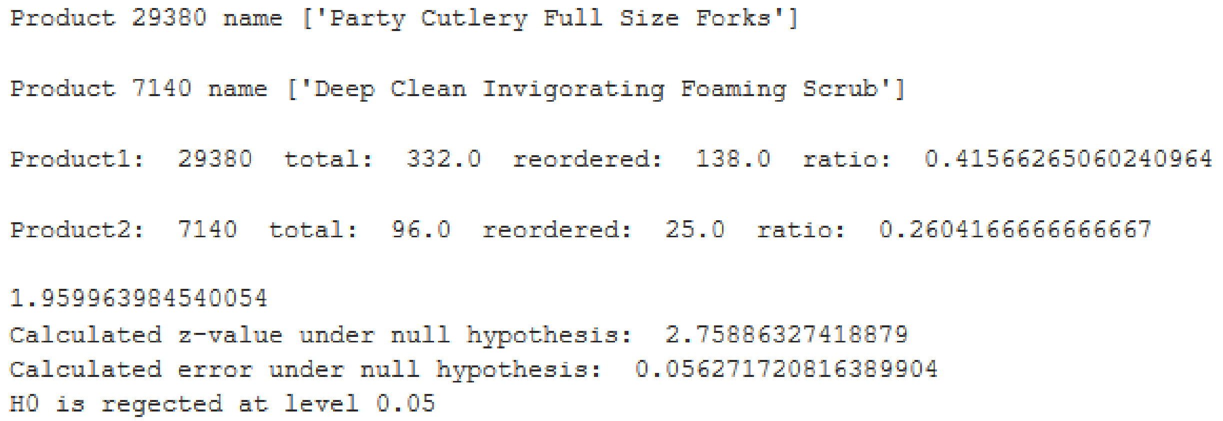
Reorder ratio varies with product also. We checked the level at which two different products with different ratios will be statistically significant. When the difference of reorder proportion is about 0.15, the products most likely belong to different populations. At lower levels, we failed to reject the hypothesis that the proportions belong to the same population.

Take for example for products numbers 25015 (Raw Fit High Protein Vanilla Single Packet) and 25094 (Brown Sugar Blend) with corresponding total orders of 66 and 124, and reorders of 14 and 39. The reorder ratios are 0.212 and 0.314. We calculated that z-value is -1.498. The value to reject hypothesis is 1.960, so we fail to reject null hypothesis. We can say that the two products are two different samples of the same population (our null hypothesis from above).

For the analysis, we wrote a function that will take two products and calculate the z-value under the null hypothesis. The function will reject the hypothesis, if z-value is higher than a given value or fail to reject. Also, the function will print information about the two products.

We also checked to see if the variables hour of the day and day of week will create distributions with different mean. To check it, we used a version of the function, that will accept as parameters two different days and times.

In most cases, we rejected the hypothesis that the reorder ratio proportion for two days and times belongs to the same set.



```
Product 29380 name ['Party Cutlery Full Size Forks']

Product 7140 name ['Deep Clean Invigorating Foaming Scrub']

Product1:  29380  total:  332.0  reordered:  138.0  ratio:  0.41566265060240964

Product2:  7140  total:  96.0  reordered:  25.0  ratio:  0.2604166666666667

1.959963984540054
Calculated z-value under null hypothesis:  2.75886327418879
Calculated error under null hypothesis:  0.056271720816389904
H0 is regected at level 0.05
```

One of the calculations

The reorder proportion by department varies greatly. We check for statistical significance. We choose two departments, with small number of items, and use the function to check if they belong to different populations. The hypothesis that the two department populations are from the same population is rejected.

We used two departments with small number of orders: “other” and “missing”. Other has total of 38086 orders and 15503 reorders, missing – 77396 orders and 30519 reorders. The reorder ratios are 0.407 and 0.394. Still, the null hypothesis is rejected. The calculated z-value is 4.12, which is far greater than the cutoff value of 1.96 and the much stricter value of 2.58.

Additional notes

1. The data collected contains between 4 and 100 orders for each user. The last order for each user is either “train” or “test”. The rest of the orders are in “prior” category.

We investigating to which categories each user belongs. When counted, the total number of users and the total number of users in prior category is 206,209. The number of users in train is 131,209 and the number of users in test is 75,000. The number of the train and test adds up to 206,209. We can conclude that there is no overlap between the two sets, because the number of users in the two add up exactly to the total number of users in the two sets.

2. Some products were not ordered many times. We can apply Central Limit Theorem if the number of times ordered is more than 50, and success and failure to reorder is at least 5. There are about 38,000 such products.

3. With tens of thousands or thousands of orders, even small variation of the reorder proportion is statistically significant.

We used z-value to determine if two different product categories are different, or the shown difference is due by chance. In most cases, we used value of 0.05 for our tests.

Machine learning

We created three models using kmeans. The variables in the first model (Model 1) are only the aisles in the data. The second model (Model 2) adds up time data and average number of items in each aisle per user. The third model (Model 2a)variables are variables of first model and max number of items.

All three model use normalized averages. We took into account that some users have more orders in the data and other have less. We divide the number of items in each aisle by the total number of orders each user has. This way, the number of orders does not skew data.

Model 1

To optimize the number of clusters we used the inertia elbow method. The graph has an elbow when we have only two cluster. The two clusters differ by how many items on average are placed from each aisle. The first ten items in both are almost identical and in the same order. The number of items cluster 0 order is

Model 2

Model two graph suggests 4 cluster according to the elbow method. The first ten items in all clusters are almost identical. What is different, is how many items on average are in each order. The clusters differ by how many items on average per aisle each order has. There are about 5.5, 12, 19.8 and 8.3 items in basked for clusters 0 – 3. This observation prompted Model 2a. Is the most important variable the number of items?

Model 2a

Adding only max number of items in basked to Model 1 gives us another model for which the elbow method suggest only 3 or 4 clusters. Both divisions of customers in clusters are similar to what we see in Model 1.

We therefore need all variables in Model 2 to divide customers in reasonable groups.

In each of the clusters there are aisles from which the contribution to the basked is almost 0. We considered threshold of 0.005 items on average per basket. The Model 2 suggest that 4 of the aisles are almost never shopped from: 'baby accessories', 'beauty', 'eye ear care', 'frozen juice', 'kitchen supplies' and 'specialty wines champagnes'. Looking at the data as whole, there are 14 aisles that have less than 0.005 items in basked. However, placing the customers in groups, only 6 appear to be lower than that threshold for all clusters.

The second ten items start showing differences in aisles. Only cluster 2 has baby food formula. Cluster 3 has bread aisle as large. It does not feature in the group frozen produce. Cluster 0 does not have lunch meat, and it is the only cluster with soft drinks in that range bracket. When we look at the second ten most ordered aisles we start seeing the difference in the clustering.

Conclusion

Customers can be divided into four main groups. Groups differ by number of items in basket, how often they shop, and the ranking by how many items in particular the customers order from a single aisle. The first ten or so in each cluster are similar. The first 15 items in all four clusters contain about the half of all ordered items.

The similar model, run on individual items fail. Even with reduction of the number of users to 5% of the original database, the model used all available memory and failed. In the future, more RAM may solve some of the problems.

Some questions asked in the beginning need further refining of the model, so the model can improve. One way to improve is to run separate models using only single cluster, or running models only on items of the middle tier, the one we actually see difference in ordering ratio.

The time for ordering is in the essence of the models discussed. We can use transformations to simplify the model by using single variable for day and time.